TEMA 6 ARRAYS: VECTORS i MATRIUS

BASE TEÒRICA

- **Concepte**

Un vector o "array" ("arreglo" en algunes traduccions) és una seqüència de objectes del mateix tipus (integer, double, char, etc), emmagatzemats un rere l'altre a la memòria i d'accés aleatori. El tipus de objecte emmagatzemat al vector pot ser qualsevol tipus de dada definida en Java. En definitiva un array és una estructura de dades que conté una col·lecció de dades del mateix tipus. Podem tenir així un vector amb les notes d'un grups d'alumnes o amb els saldos d'un conjunt de clients.

Normalment se li diu vector quan tenim un array d'una dimensió i se li diu matriu si és de dues dimensions. A destacar que es poden tractar arrays de més de dues dimensions.

- **Declaració, creació i inicialització de vectors**

Un vector es declara segons la sintaxi:

tipus_dada[ ] identificador_array;

o també: ***tipus_dada identificador_array[ ];***

Per exemple un vector de nombres reals tipus *float* i de nom caixa es declara com:

float [] caixa; o també: float caixa [];

Un cop declarat un vector s'ha de crear, de manera que es reservi en memòria l'espai necessari per a emmagatzemar tots els seus elements. Per fer això haurem d'utilitzar l'operador **new()** indicant el nombre d'elements que tindrà el vector (dimensió del vector).

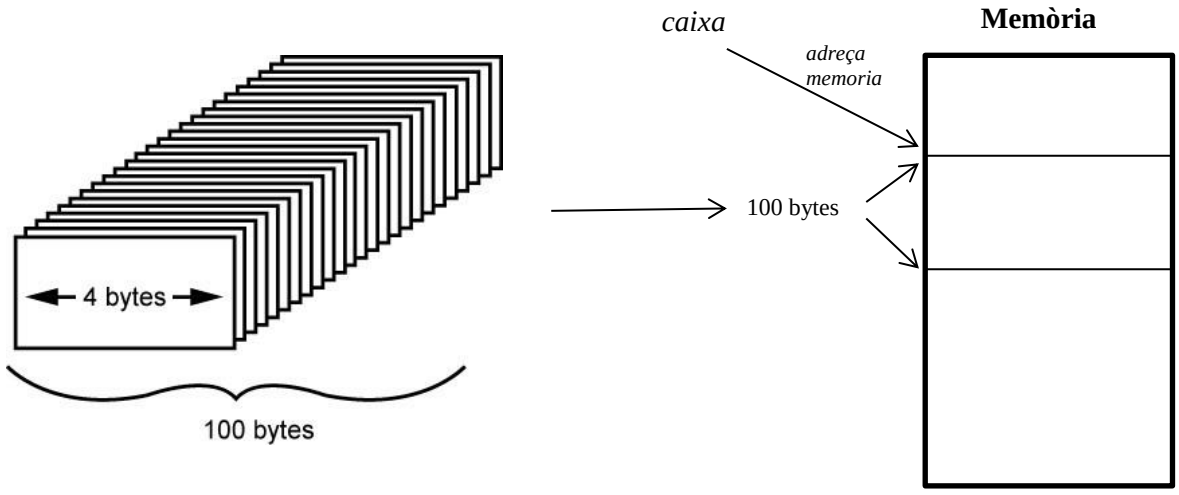
identificador_array = new tipus_dada[nombre_elements];

Per exemple: *caixa = new float [25];*

Es pot declarar i crear un array al mateix temps: *float [] caixa = new float [25];*

Com cada *float* n'ocupa 4 bytes, i en tenim 25 elements, el vector *caixa* ocuparà 100 bytes. Aquests 100 bytes estaran col·locats a la memòria a partir de l'adreça de memòria o referència (localització del contingutdo de l'array a la memòria) indicada pel nom del vector, en aquest cas: *caixa.*

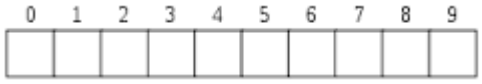
float [] caixa = new float [25];



Per exemple: *double [ ] notes= new double[10];*

és la declaració i creació de la variable *notes* que identifica un vector on pot haver-hi fins a 10 valors reals. Gràficament es pot representar com:

notes:



Per accedir a una casella del vector hem de fer servir el mètode de la indexació, consistent en fer servir un nombre enter anomenat subíndex, que indica la posició de l'element dins del vector. En Java els elements comencen en el 0. Així,el primer element del vector **notes** serà **notes[0]**, el segon **notes[1]**, etc., i l'últim**notes[9].**

notes[0]	notes[1]	notes[2]	notes[3]	notes[4]	notes[5]	notes[6]	notes[7]	notes[8]	notes[9]
7	-5	10	3	4.2	7	8	-3	6.5	5

Com tota variable, un vector pot tenir valors inicials després de la seva declaració. La sintaxi per declarar i inicialitzar un vector és:

tipus_dada identificador_array [ ] = { llista_de_valors separats per comes };

La llista de valors és una llista separada per comes ", " de valors constants que son del mateix tipus que el tipus base del vector.

Exemple: *String dies[] = {"Dil", "Dim", "Dix", "Dij", "Div", "Dis", "Diu"};*

int num[ ] = {0,1,2,3,4};

En aquest exemple la primera constant de valor 0 s'emmagatzema a la primera posició del vector, la segona constant de valor 1 a la segona posició del vector, i així successivament fins completar les cinc constants. El compilador reservarà l'espai necessari de memòria i situarà les dades en posicions contigües. Noteu que no es declara la mida del vector. El compilador la determina automàticament calculant el nombre de valors enumerats.

D'aquesta manera l'assignació següent*: int num[] = {0,1,2,3,4};* fa que la dimensió del vector *num* sigui 5.
Si un array no s'inicialitza, el compilador inicialitzarà per defecte tots els seus elements a 0, o en general a l'element base del tipus de dada.
En general: Array de dimensió (longitud, nombre d'elements): **n**
Rang d'un array: **0 .. n-1**
Accès element ièssim: **nom_array[i]**
En array sempre podrem sabe el nombre d'elements màxim que podrà contenir (dimensió) amb el mètode **length**:

nom_array.length

Es molt recomanable declarar les dimensions dels arrays utilitzant constants per evitar així modificacions (un cop s'ha creat un array no es pot modificar la seva dimensió).

```
final int ARRAY_SIZE = 100;      //declara constant ARRAY_SIZE
...
int notes[] = new int[ARRAY_SIZE];
```

- **Declaració, creació i inicialització de matrius**

En el cas de una matriu (vector de dos dimensions) es declara fent una extensió per la segona dimensió d'un vector:

tipus_dada[ ] [ ] identificador_array;

o també: ***tipus_dada identificador_array[ ] [ ];***

on la primera dimensió es correspon a les files i la segona a les columnes començant sempre des de zero.
Per crear una matriu haurem de fer l'extensió corresponent al cas d'un vector però amb dues dimensions:

identificador_array = new tipus_dada[nombre_files] [nombre_columnes];

Per exemple la matriu: *alumnes[4][6]* representada gràficament serà:

	0	1	2	3	4	5
0						
1						
2						
3						
4						

Pel cas de posar valors inicials als vectors multidimensionals o matrius podem fer:

int tab[2][5] = {{0,1,2,3,4},{5,6,7,8,9}};

equivalent a: *int tab[2][5] = {0,1,2,3,4,5,6,7,8,9};*

on es declara una matriu d'enters de dues files i cinc columnes.

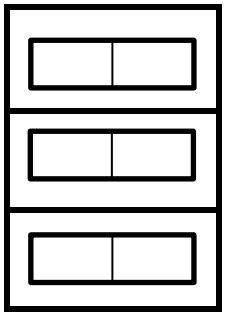
Serà molt habitual l'ús de constants a l'hora de crear una matriu:

```
const int FILS = 3;
const int COLS = 2;
int[][] matriu = new int[FILS][COLS];
```

També podem utilitzar el mètode length, saben que:

matriu.length : nombre de files
matriu[0].length : nombre de columnes

ja que per Java, una matriu és un vector de vectors. Per exemple, en el cas anterior matriu és un vector de 3 files i a cada fila hi ha un altre vector de 2 caselles:



- **Operacions amb arrays**

Les operacions que usualment es realitzaran amb arrays durant el procés de resolució d'un problema son:

- *Accés als elements de l'array.*

S'accedeix a cada component d'una variable array usant el seu nom seguit del que s'anomena **índex** entre [ ]. L'índex d'un element d'un array és un valor enter que indica l'ordre de l'element dins de l'array. Els ***índexs començaran sempre per 0***, es a dir el primer element d'un array té índex 0. Així doncs, un element situat a la posició i, en realitat estarà situat en l'ordre i+1.

Vector[i] → element a la posició i

Matriu[i][j] → element a la fila i, columna j de Matriu

- *Recorregut (accés seqüencial)*

A l'operació d'efectuar una acció general sobre tots els elements d'un vector s'anomena recorregut del vector. Aquestes operacions es realitzaran utilitzant normalment l'estructura for i utilitzant la variable de control (per exemple , i) com índex de l'array (per exemple, matriu[i]). L'increment del comptador del bucle produirà el tractament successiu dels elements del vector.

En general, les operacions sobre els arrays impliquen el processament o tractament dels elements individuals de l'array.

- *Assignació.*

S'assigna un valor a un array, assignant valors a les seves components individuals. Haurem d'assignar els valors element a element, mitjançant una estructura iterativa, normalment un for:

```
int[] notes = new int[10];
for (int i = 0; i < notes.length; i++) {
        notes[i] = -1;
}
```

- *Igualtat.*

La única forma de poder verificar la igualtat de dos arrays es realitzant una estructura iterativa que compari tots i cadascú dels elements dels arrays que tinguin el mateix valor de índex.

- *Lectura/escriptura.*

No està permesa la lectura i escriptura directa d'arrays. Aquest tipus d'operacions s'ha de realitzar element a element i mitjançant estructures repetitives, normalment un for:

Lectura d'un array;

```
Scanner input = new Scanner(System.in);
for (i = 0; i < MAXIM; i++) {
        notes[i] = input.nextDouble();
}
```

Escriptura d'un array:

```
for (i = 0; i < MAXIM; i++) {
        System.out.println(notes[i]);
}
```

- *Còpia.*

Si donats un arrays v1 el volem copiar i fem una assignació tipus:

```
int [] v2 = v1;     //Això no fa una copia dels vectors!!!!
```

No crearem un nou array si no que el que farem serà copiar la referència (adreça de memòria) de l'array v1 a v2. Per fer una copia, hem de posar els elements un a un del vector v1 al vector v2:

```
int[] v2 = new int[v1.length];
```

```
for (i=0; i < v1.length; i++)  v2[i] = v1[i];
```

- **Us amb funcions (mètodes)**

Es poden usar arrays com a paràmetres de mètodes i com a valor de retorn dels mètodes. Hem de tenir en compte que els arrays en Java sempre es passen per referència als mètodes, es a dir el mètode treballa amb una copia de la localització de l'array i no amb una copia del seu contingut (conjunt dels seus valors). Això suposa que podem modificar els valors d'un array dins d'un mètode i els canvis es mantindran a partir d'aquest moment al llarg del programa.

Per exemple, podem fer servir un mètode per llegir les dades d'un vector i omplir-lo, i el seu contingut es mantindrà fora de l'àmbit del mètode:

```
public static void llegiVector(int[] v1) {
        for (int i = 0; i < v1.length; i++) {
                System.out.println("Introudiu una dada");
                v[i] = input.nextInt();
        }
}
```

Altres exemples:

- Mètode per sumar els elements d'un array:

```
public static double sum(double[] data) {
        double total = 0;
        for (int i = 0; i < data.length; i++) {
                total = total + data[i];
        }
        return total;
}
```

- Mètode per calcular la mitjana dels elements d'un vector:

```
public static double mitjana( int [] v) {
        int suma = 0;
        for( int i = 0; i < v.length; ++i )
                suma = suma + v[ i ];
        return suma/ v.length;
}
```

On la crida des de el main serà: *sum(notes);* i *mitjana(notes);*

- Mètode per llegir els elements d'una matriu:

```
public static void llegir_matriu(double [][] m){
        for ( int i = 0; i < m.length; ++i){
                for ( int j = 0; j < m[0].length; ++j)
                        m[i] [j] = input.nextDouble();
                }
        }
}
```

La crida des de el main serà: *matriu(notes);*

- **La classe Array**

La biblioteca de classes de Java inclou la classe:

java.util.Arrays

que proporciona mètodes molt útils per tasques molt comunes a l'hora de treballar amb vectors i matrius:

- *Arrays.sort(v)*: ordena els elements d'un vector v.
- *Arrays.equals(v1,v2)*: comprova si dos vectors v1 i v2 son iguals.
- *Arrays.fill(v,val)*: omple el vector v amb el valor val.
- *Arrays.toString(v)*: retorna un string que representa el contingut del vector v.
- *Arrays.binarySearch(v, k)*: busca el valor k dins del vector v (previament ordenat)

També disposem d'un mètode per copiar arrays de la biblioteca estàndar de Java:

System.arraycopy(arrayorigen,indexorigen,arraydesti,indexdesti,longitud);

BASE PRACTICA

- **Programes d'exemple.**

1.- Editar, compilar i provar que el següent programa fa la mitjana de tres notes tractades amb un vector

```
package mitjananotes;
import java.util.Scanner;
public class MitjanaNotes {
        public static void main(String[] args) {
                double[] notes;                 //declaració variable vector notes
                notes = new double[3];          //es monta a memoria el vector de 3 caselles
                double mitjana;
                Scanner teclado = new Scanner(System.in);
                System.out.println("Introdueix una nota: ");
                notes[0] = teclado.nextDouble();
                System.out.println("Introdueix una nota: ");
                notes[1] = teclado.nextDouble();
                System.out.println("Introdueix una nota: ");
                notes[2] = teclado.nextDouble();
                mitjana = (notes[0]+notes[1]+notes[2])/3;
                System.out.println("La mitjana es: " +   mitjana);

}
```

2.- Editar, compilar i provar que el següent programa fa la mitjana de n notes d'un màxim de MAX tractades amb un vector. També fa diversos càlculs estadístics.

```
package mitjananotes;
import java.util.Scanner;
public class MitjanaNotes {
        public static void main(String[] args) {
                final int MAX = 100;                    //constant pel nombre de notes
                double[] notes = new double[MAX];
                double suma = 0;                    //Calcula la suma de notes
                int n;                              //nombre de notes reals
                int aprovats = 0, suspesos = 0, novalid = 0;
                double max = 0, min = 0;
                Scanner teclado = new Scanner(System.in);
                System.out.println("Introduiu el nombre de notes del curs (Maxim " + MAX + ") : ");
                n = teclado.nextInt();
                //Introduccio de notes
                for(int i = 0; i < n; i++) {
                   System.out.println("Introduiu una nota: ");
                   notes[i] = teclado.nextDouble();
                 }
                //Mostrem el vector
                System.out.println("Les notes introduides: ");
                for(int i = 0; i < n; i++) {
                System.out.println(notes[i]);
                }
                //Mostra les notes per patalla
                for(int i = 0; i < n; i++) {
                   if(i == 0){
                      System.out.print("(" + notes[i]);
                   }
                   else if(i == n-1){
                      System.out.println("," + notes[i] + ")");
                   }else{
                      System.out.print("," + notes[i]);
                   }
                }
                //Calcul
                for(int i = 0; i < n; i++) suma = suma + notes[i];
                //System.out.println("El vector es: " + notes);     //Això és un error
                System.out.println("El resultat de la mitjana es: " +   suma/n);
                for(int i = 0; i < n; i++){
                   if(notes[i] >= 0 && notes[i] < 5) suspesos++;
                   else  if(notes[i] >= 5 && notes[i] <= 10) aprovats++;
                   else novalid++;
                }
                System.out.println("Nombre d'aprovats: " +  aprovats);
                System.out.println("Nombre d'aprovats: " +  (aprovats/n)*100);
                System.out.println("Nombre de suspesos: " +  suspesos);
                System.out.println("Nombre de suspesos: " +  (suspesos/n)*100);
                for(int i = 0; i < n; i++) {
                   if(notes[i] > max) max = notes[i];
                   if(notes[i] < min) min = notes[i];
                }
                System.out.println("El maxim es " +  max);
                System.out.println("El minim es " +  min);
```

```
        }
}
```

3.- Editar, compilar i provar que el següent programa fa el producte escalar de dos vectors.

```
package prodesc;
import java.util.Scanner;

public class Productescalar {
        public static void main(String[] args) {
                final int MAX = 5;
                double[] v1 = new double[MAX];
                double[] v2 = new double[MAX];
                double suma = 0;                    //Calcula la suma
                int n;                              //nombre de compnents reals

                Scanner teclado = new Scanner(System.in);
                System.out.println("Nombre de components dels vectors (Maxim " + MAX + ") : ");
                n = teclado.nextInt();

                //Introduccio dels elements de cada vector
                for(int i = 0; i < n; i++) {
                    System.out.println("Introduiu una dada ");
                    v1[i] = teclado.nextDouble();
                }
                //Llegim el vector v2
                for(int i = 0; i < n; i++) {
                    System.out.println("Introduiu una dada ");
                    v2[i] = teclado.nextDouble();
                }
                //Mostra el vector v1
                for(int i = 0; i < n; i++){
                    if(i == 0){
                       System.out.print("(" + v1[i]);
                    }
                    else if(i == n-1){
                       System.out.println("," + v1[i] + ")");
                    }else{
                       System.out.print("," + v1[i]);
                    }
                }
                //Mostra el vector v2
                for(int i = 0; i < n; i++){
                   if(i == 0){
                      System.out.print("(" + v2[i]);
                   }
                   else if(i == n-1){
                      System.out.println("," + v2[i] + ")");
                   }else{
                      System.out.print("," + v2[i]);
                   }
                }
                //Calcul del producte escalar
                for(int i = 0; i < n; i++) {
                  suma = suma + v1[i]*v2[i];
                }
                //Mostrem el resultat
                System.out.println("El resultat del producte escala es: " + suma );
            }
}
```

4. - Editar, compilar i provar el programa anterior però utilitzant funcions.

```
package producteescalar;
import java.util.Scanner;
public class ProducteEscalar {
   public static void main(String[] args) {
      final int MAX = 5;
      double[] v1 = new double[MAX];
      double[] v2 = new double[MAX]
      int n;                                  //nombre de compnents reals
      Scanner teclado = new Scanner(System.in);
      System.out.println("Nombre de components dels vectors (Maxim " + MAX + ") : ");
      n = teclado.nextInt();
      OmplirVector(v1,n);    //Lectura del vector v1
      OmplirVector(v2,n);    //Llegim el vector v2
      MostraVector(v1, n); //Mostra el vector v1
      MostraVector(v2, n); //Mostra el vector v2
      ProdEscalar( n, v1, v2);       //Producte escalar
   }
   public static void MostraVector (double v[ ], int k) {
            for(int i = 0; i < k; i++) {
         if(i == 0){
            System.out.print("(" + v[i]);
         } else if(i == k-1) {
            System.out.println("," + v[i] + ")");
         } else {
            System.out.print("," + v[i]);
         }
      }
   }
   public static void OmplirVector (double v[ ], int k) {
            Scanner teclado = new Scanner(System.in);
      for(int i = 0; i < k; i++) {
         System.out.println("Introduiu una dada ");
         v[i] = teclado.nextDouble();
      }
}
public static void ProdEscalar(int k, double v1[ ], double v2[ ]) {
     double suma = 0;
     for(int i = 0; i < k; i++) {
        suma = suma + v1[i]*v2[i];
     }
```

```
    System.out.println("El resultat del producte escalar es: " + suma );
  }
}
```

5.- Editar, compilar i provar que el següent programa fa la mitjana de unes notes fent servir vectors i funcions.

```
package vectormitjanotes;
import java.util.Scanner;
public class VectorMitjaNotes {
    public static void main(String[] args) {
        final int MAX = 5;
        double[] notas = new double[MAX];
        llegirVector(notas);
        mostraVector(notas);
        System.out.println("La mitajan de les notes es: " + mitjana(notas));
        System.out.println("La nota maxima es " + maxim(notas));
        System.out.println("La nota minima es " + minin(notas));
    }
    public static void llegirVector (double[] notas) {
         Scanner input = new Scanner(System.in);
        System.out.println("Lectura de los elementos del array: ");
        for (int i = 0; i < notas.length; i++) {
            System.out.print("Intoduir la nota " + i + ":") ;
            notas[i] = input.nextDouble();
        }
    }
    public static void mostraVector (double[] notas) {
        System.out.println("Lectura de los elementos del array: ");
        for (int i = 0; i < notas.length; i++) {
            if (i == 0) System.out.print("( " + notas[0] + ",");
            else if (i == notas.length-1) System.out.println(notas[notas.length-1] + " )");
            else System.out.print(" " + notas[i] + " ,");
        }
    }
    public static double mitjana (double [] notas) {
        double sum = 0;
        for (int i = 0; i < notas.length; i++) {
            sum = sum + notas[i];
        }
        return sum/notas.length;
    }
    public static double maxim (double [] notas) {
        double max = notas[0];
        for (int i = 1; i < notas.length; i++) {
            if(notas[i] > max) max = notas[i];
        }
        return max;
    }
    public static double minin (double [] notas) {
        double min = notas[0];
        for (int i = 1; i < notas.length; i++) {
            if(notas[i] < min) min = notas[i];
        }
        return min;
```

```
    }
 }
```

6.- Editar, compilar i provar el següent programa que treballa amb una matriu.

```
import java.util.Scanner;
public class Matriu {
   public static void main(String[] args) {
      final int FIL = 5, COL = 4;
      Scanner sc = new Scanner(System.in);
      int i, j, max, min;
      int filaMajor, filaMenor, colMajor, colMenor;
      int[][] A = new int[FIL][COL];            //Es crea una matriu de 5 files i 4 columnes
      System.out.println("Dades de la matriu: ");
      for (i = 0; i < FIL; i++) {
         for (j = 0; j < COL; j++) {
            System.out.print("A[" + i + "][" + j + "]= ");
            A[i][j] = sc.nextInt();
         }
      }
      System.out.println("Valors introduïts a la matriu:");
      for (i = 0; i < A.length; i++) {
         for (j = 0; j < A[0].length; j++) {
            System.out.print(A[i][j] + " ");
         }
         System.out.println();
      }
      max = min = A[0][0];       // primer element de la matriu com a max i min
      filaMajor = filaMenor = colMajor = colMenor = 0;
      for (i = 0; i < A.length; i++) {  //per a cada fila de la matriu
         for (j = 0; j < A[0].length; j++) { //per a cada columna de la matriu
            if (A[i][j] > max) {
               max = A[i][j];
               filaMajor = i;
               colMajor = j;
            } else if (A[i][j] < min) {
               min = A[i][j];
               filaMenor = i;
               colMenor = j;
            }
         }
      }
      System.out.print("Element maxim: " + max);
      System.out.println(" Fila: "+ filaMajor + " Columna: " + colMajor);
      System.out.print("Element minim: " + min);
```

```
      System.out.println(" Fila: "+ filaMenor + " Columna: " + colMenor);
   }
}
```

7.- Provar el següent programa que treballa amb una matriu fent servir funcions.

```
import java.util.Random;
import java.util.Scanner;
public class Matrius {
   public static void main(String[] args) {
      final int F = 3;
      final int C = 3;
      int[][] m = new int[F][C];
      int valor;
      inicializar(m);
      System.out.println("Matriu inicialitzada");
      mostrar(m);
      llegir(m);
      System.out.println("Matriu llegida");
      mostrar(m);
      valor = demanaValor(m);
      System.out.println("L'element demanat es: " + valor);
   }
   public static void llegir(int[][] matriu){
      Scanner teclat = new Scanner(System.in);
      for(int i=0;i<matriu.length;i++){
         for(int j=0;j<matriu[i].length;j++){
               System.out.println("Fila "+ i + " columna "+ j +":");
               matriu[i][j] = teclat.nextInt();
         }
      }
   }
   public static void inicializar(int[][] matriu){
      Random r = new Random();
      for(int i=0;i<matriu.length;i++){
         for(int j=0;j<matriu[i].length;j++){
            matriu[i][j] = r.nextInt(10) + 1; //entre 1 i 10
         }
      }
   }
   public static void mostrar(int[][] matriu){
      for(int i=0;i<matriu.length;i++){
         for(int j=0;j<matriu[i].length;j++){
            System.out.print(matriu[i][j] + "\t");
         }
         System.out.println();
      }
   }
   public static int demanaValor(int[][] matriu){
      Scanner teclat = new Scanner(System.in);
      int fil = 0,col = 0;
      System.out.println("Numero de fila:");
      fil = teclat.nextInt();
      System.out.println("Numero de columna:");
      col = teclat.nextInt();
```

```
    return matriu[fil-1][col-1];
  }
}
```

8.- Dissenyar un programa per calcular la matriu simètrica d'una matriu NxN

```
import java.util.Scanner;

public class MatriuSim {
  public static void main(String[] args) {
    final int N = 100;
    int dim;
    double[][] m = new double[N][N];
    Scanner teclat = new Scanner(System.in);
    System.out.println("Posa la dimensio de la matriu (maxim " + N +" :");
    dim = teclat.nextInt();
    if (dim <= N) {
                System.out.println("Introdueix els elements de la matriu: ");
                llegirMatriu(dim,m);
                System.out.println("La matriu: ");
                mostrarMatriu(dim,m);
                if (simetrica(dim,m)) {
                        System.out.println("Es simetrica");
                } else {
                        System.out.println("NO es simetrica");
                }
        } else {
      System.out.println("Dimensio incorrecta.");
    }
  }
  public static void llegirMatriu(int dim, double[][] matriu){
    Scanner teclat = new Scanner(System.in);
    for(int i=0;i<dim;i++){
      for(int j=0;j<dim;j++){
           System.out.println("Fila "+ i + " columna "+ j +":");
           matriu[i][j] = teclat.nextDouble();
      }
    }
  }
  public static void mostrarMatriu(int dim, double[][] matriu){
    for(int i=0;i<dim;i++){
      for(int j=0;j<dim;j++){
        System.out.print(matriu[i][j] + "\t");
      }
      System.out.println();
    }
  }
  public static boolean simetrica(int dim, double[][] matriu){
    int i = 0, j;
        boolean simet = true;
        while (i<dim && simet) {
                j=0;
                while (j<i && simet) {
                        if (matriu[i][j]!=matriu[j][i]) {
                                simet = false;
                        } else j++;
                }
                if (simet) i++;
        }
```

```
        return simet;
    }

}
```

- **Programes per desenvolupar**

1.- Feu un programa a partir dels programes anteriors, ompli un vector amb les temperatures diàries d'un mes (suposem de 30 dies) i ens informi del nombre de dies del mes que han tingut la seva temperatura dins d'un rang establert prèviament per l'usuari.

2.- Feu un programa a partir dels programes anteriors, ompli una matriu amb les notes trimestrals (columnes 1, 2 i 3) d'un grup de "n" alumnes d'un màxim de "NALUM" de manera aleatòria. A la columna 0 de la matriu es demanarà una possible nota extra de cada alumne. El programa ha de calcular la nota final de cada alumne com a resultat de la mitjana de les notes trimestrals més la extra, i situar-la a la columna 4 de la matriu. Finalment el programa ha de mostrar per pantalla la matriu de notes obtinguda. Opcionalment es pot demanar a l'usuari que demani les notes concretes d'un alumne a partir del seu numero.

- **Programes opcionals**

1.- Fer un programa que:

a) Llegeixi una seqüència de "n" valors numèrics reals d'un màxim de MAX i els emmagatzemi en un vector.

b) Mostri per pantalla quin és el valor mínim, així com la posició que n'ocupa en el vector. Si apareix repetit el valor mínim es mostrarà el de menor índex dels valors mínims.

2. Calculeu la matriu transposada d'una matriu. Una matriu transposada d'una donada, és el resultat de canviar en la matriu original el valor de les files pel de les columnes.

Es definirà una funció amb el prototipus:

```
public static void transposada(double m[][], double mt[][]);
```

Aquesta funció rebrà la matriu original introduïda pel teclat m, i ha de calcular la seva transposada mt. El programa principal ha de mostrar per pantalla la matriu original i la seva transposada.